



# **Сравнение четырех алгоритмов распараллеливания для неявного метода Монте-Карло с распределением решений по областям**

**Томас А. Браннер (СНЛ)  
Тодд Дж. Урбач и Томас М. Эванс (ЛАНЛ)  
Николас А. Джентайл (ЛЛНЛ)**

**Совместная российско-американская конференция пяти  
лабораторий по вычислительной математике и физике  
19-23 июня 2005 года  
Вена, Австрия**

Сандия является многопрофильной лабораторией, которая управляется Корпорацией «Сандия», принадлежащей компании «Локхид Мартин», для Администрации по национальной ядерной безопасности Минэнерго США в рамках контракта DE-AC04-94AL85000.

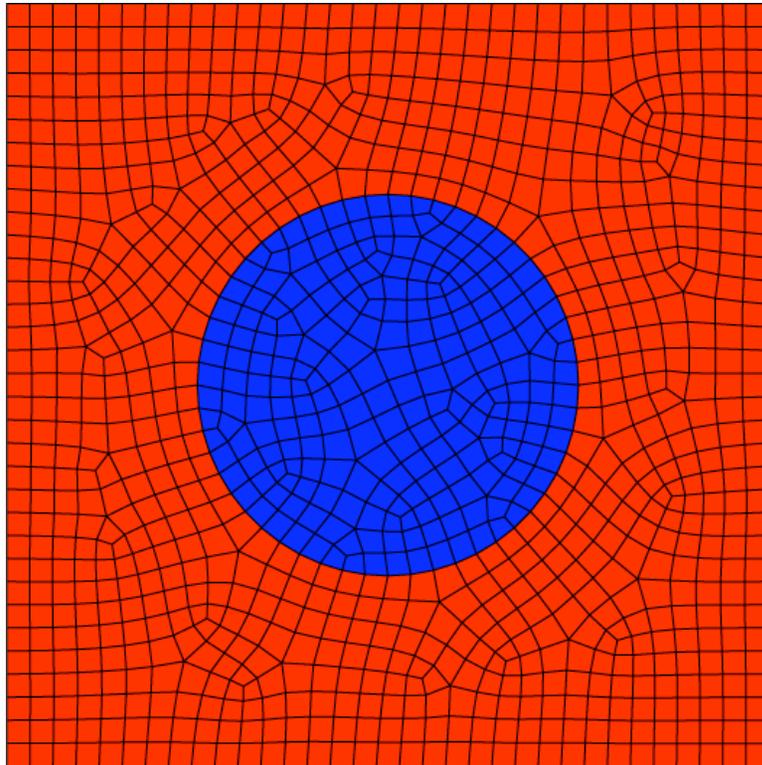
Лос-Аламосская национальная лаборатория, работодатель, предоставляющий равные возможности для всех социальных групп, управляется Университетом Калифорнии для Министерства энергетики Соединенных Штатов по контракту W-7405-ENG-36.

Эта работа выполнена под эгидой Министерства энергетики США Ливерморской национальной лабораторией им. Лоуренса, управляемой Университетом Калифорнии в рамках контракта W-7405-ENG-48.

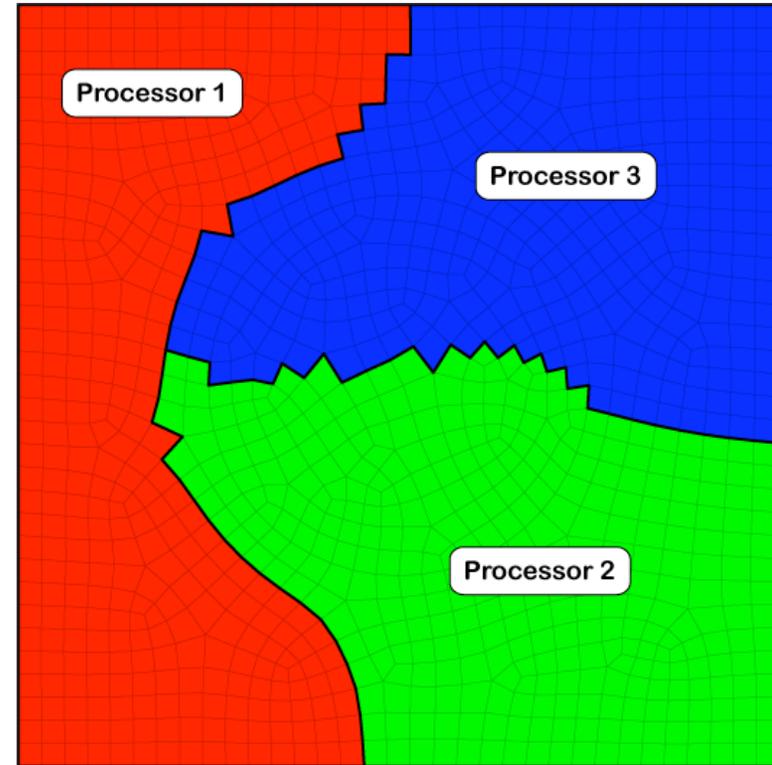




## Моделирование частиц на разложенной сетке



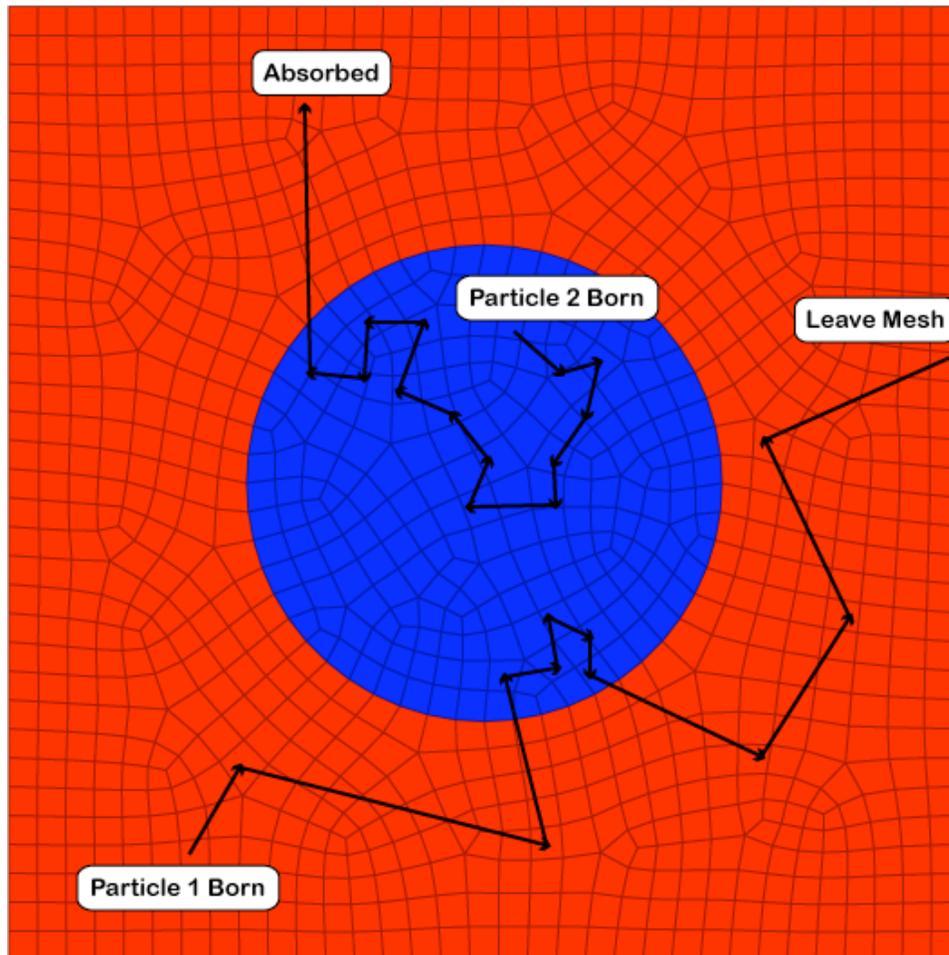
- Обычная сетка пользователя
- Цилиндр из одного материала
- Другой материал снаружи



- Зоны сбалансированной загрузки
- Обычное разложение
- Нерегулярные границы



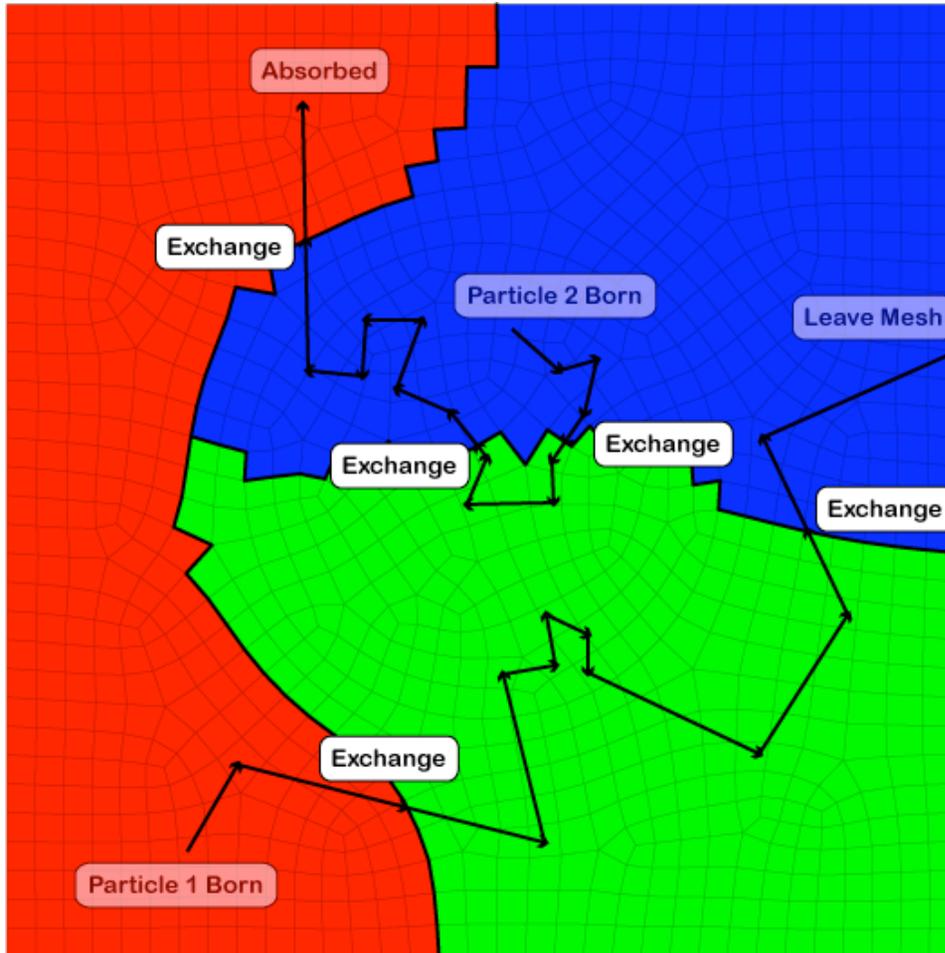
# Физическое моделирование



- Рождение частиц в сетке
- Рассеяние в материалах
- Отложение энергии в зонах
- Обсчет фотонов заканчивается, когда они:
  - уходят из сетки
  - поглощаются
  - или достигают конца временного шага.



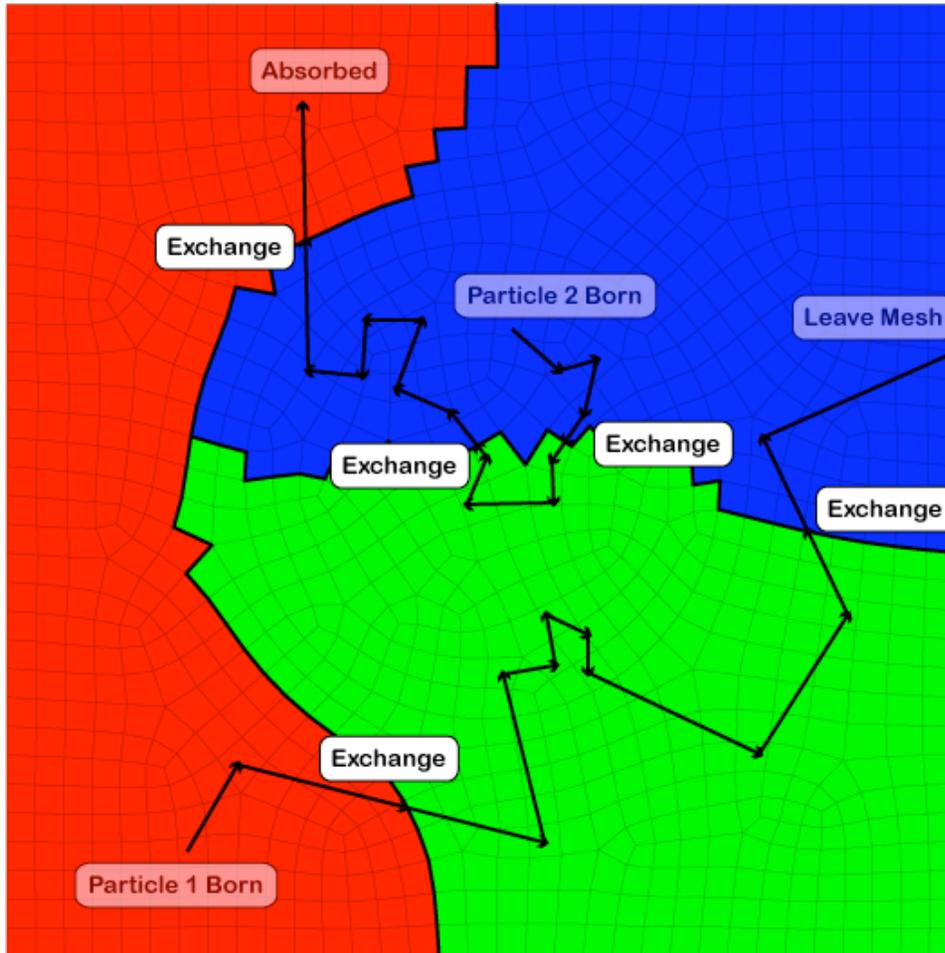
# Моделирование разложения



- Новые события в разложенной сетке:
  - Между процессорами должен произойти обмен частицами.
  - Необходимость глобального определения завершения расчета всех частиц.
- Четыре алгоритма по-разному выполняют эти два шага.



# Алгоритм 1: KULL

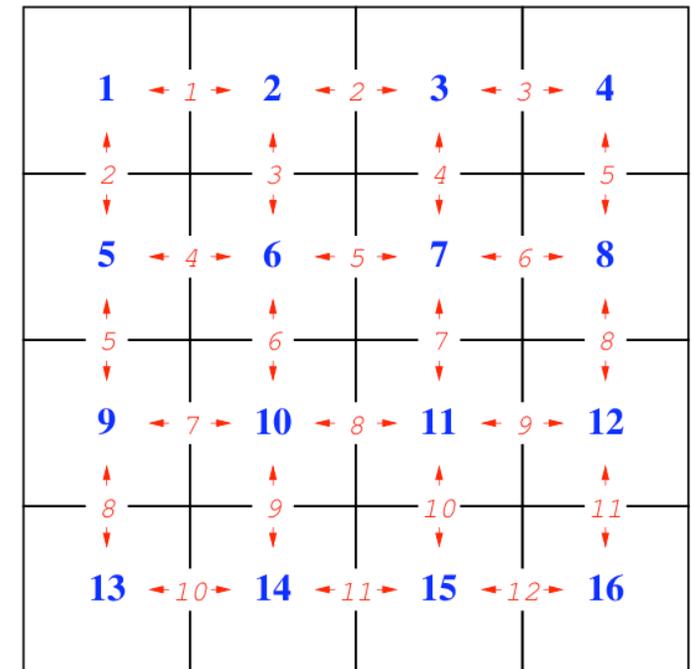


1. Каждый процессор переносит локальные частицы до тех пор, пока все они:
  - достигнут границ процессора,
  - будут поглощены,
  - выйдут за пределы сетки, или
  - достигнут конца временного шага.
2. Блокирование обмена частицами, скопившимися в буферной зоне на границах процессора, между соседними областями.
3. Глобальная сумма подсчитывает количество оставшихся частиц для моделирования на всех процессорах.
4. Если сумма частиц не равна нулю, то делается переход на шаг 1.



# Проблемы, связанные с алгоритмом 1

- Блокирование передач для обмена частицами создает последовательную форму связи.
- Многократная синхронизация точек на временном шаге для расчета глобальной суммы, которая используется для определения завершения расчета всех частиц.
- Алгоритм 2 является модификацией алгоритма 1 и позволяет использовать асинхронный обмен частицами, устраняя первую проблему.
- Алгоритм 3 устраняет синхронизацию точек, но не очень хорошо масштабируется.

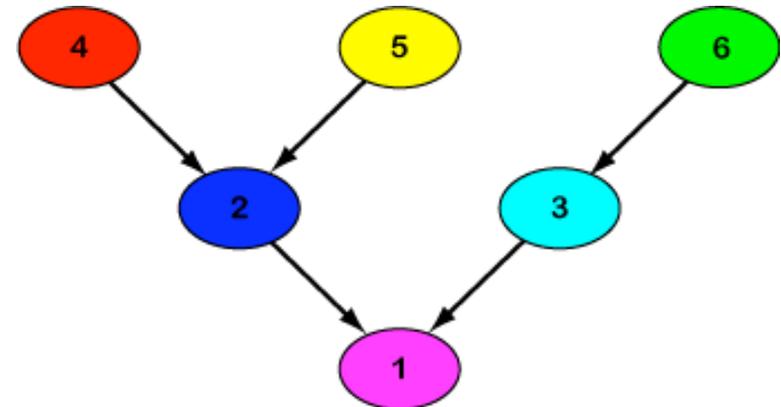
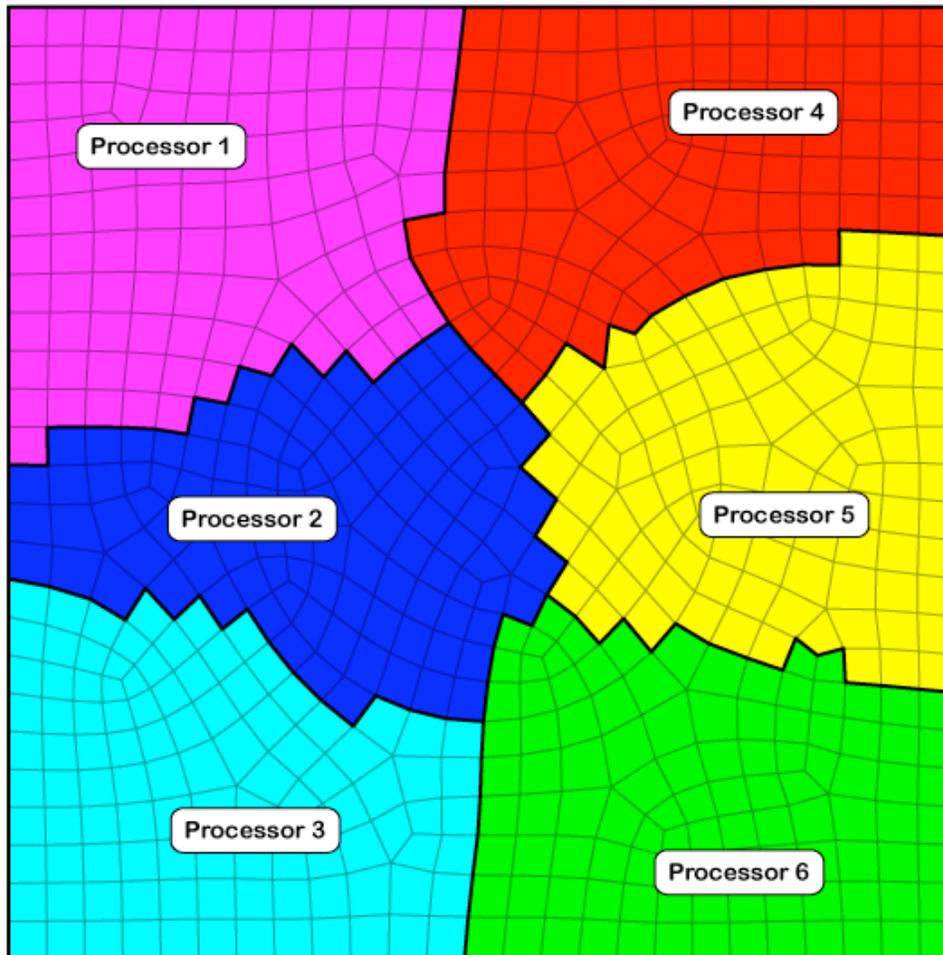


Key: **Processor/Domain Number**  
← Communication Step →

Пример разложения для иллюстрации последовательной связи в алгоритме 1.



## Алгоритм 4: улучшенный алгоритм Милагро



- Более частые асинхронные обмены частицами между буферами меньшего, фиксированного размера.
- Главный процессор (процессор 1) отслеживает общее количество посчитанных частиц
  - собранных асинхронно с использованием двоичного дерева.



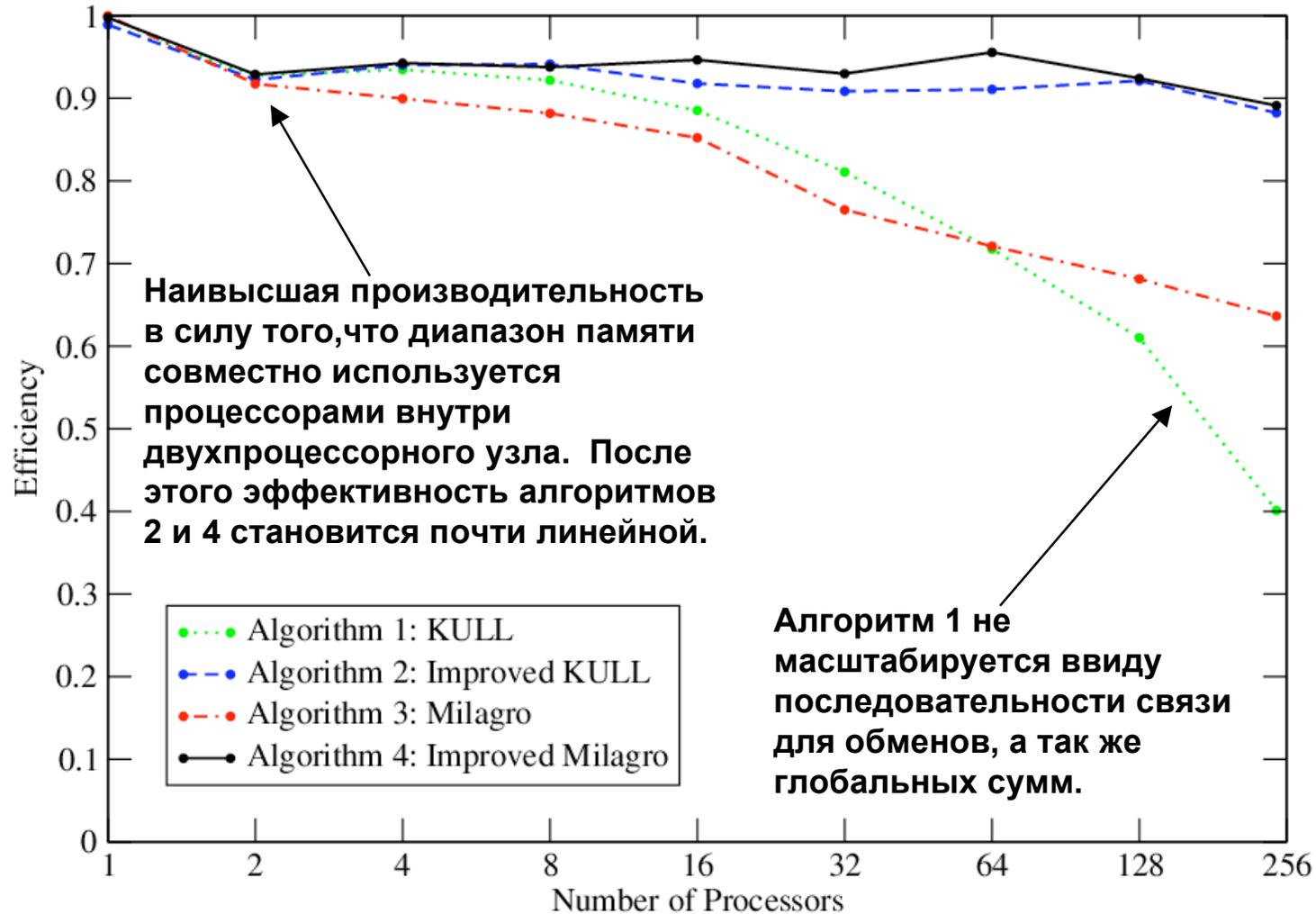
## Пробная задача 1: разогрев куба

---

- Куб с равномерной температурой
  - Зоны 60x60x60
  - Каждая зона примерно равна одной средней длине пробега
- Граничные условия отражения
- Идеально сбалансированная загрузка
- Изучение масштабирования постоянной работы (по строгим правилам)
  - Та же самая задача подразделяется на множество разложений
  - Тот же результат от всех четырех алгоритмов



# Изучение постоянной работы для разогрева куба

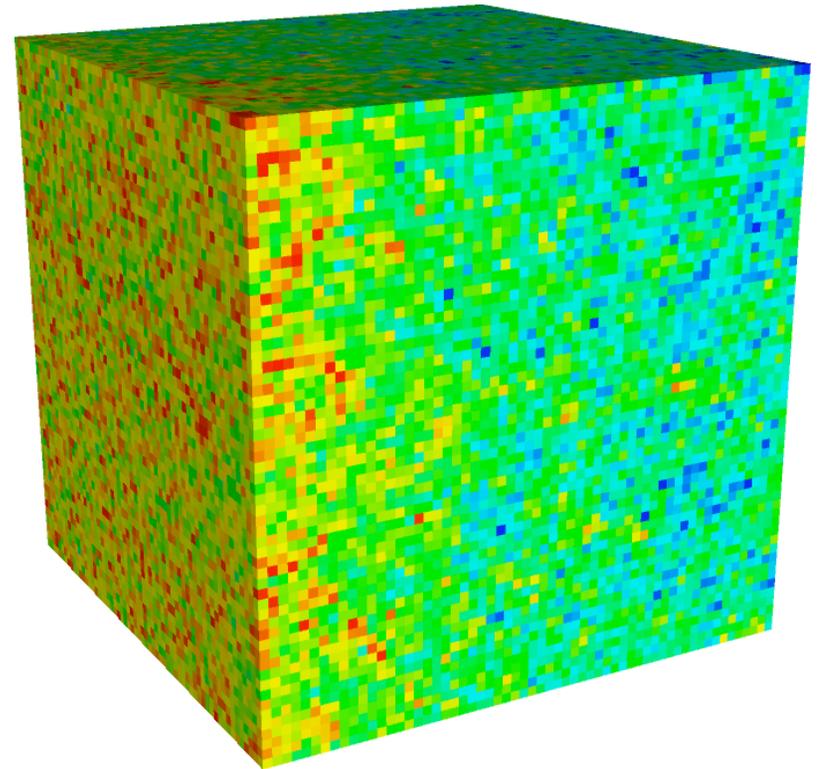




## Пробная задача 2: небольшой дисбаланс загрузки

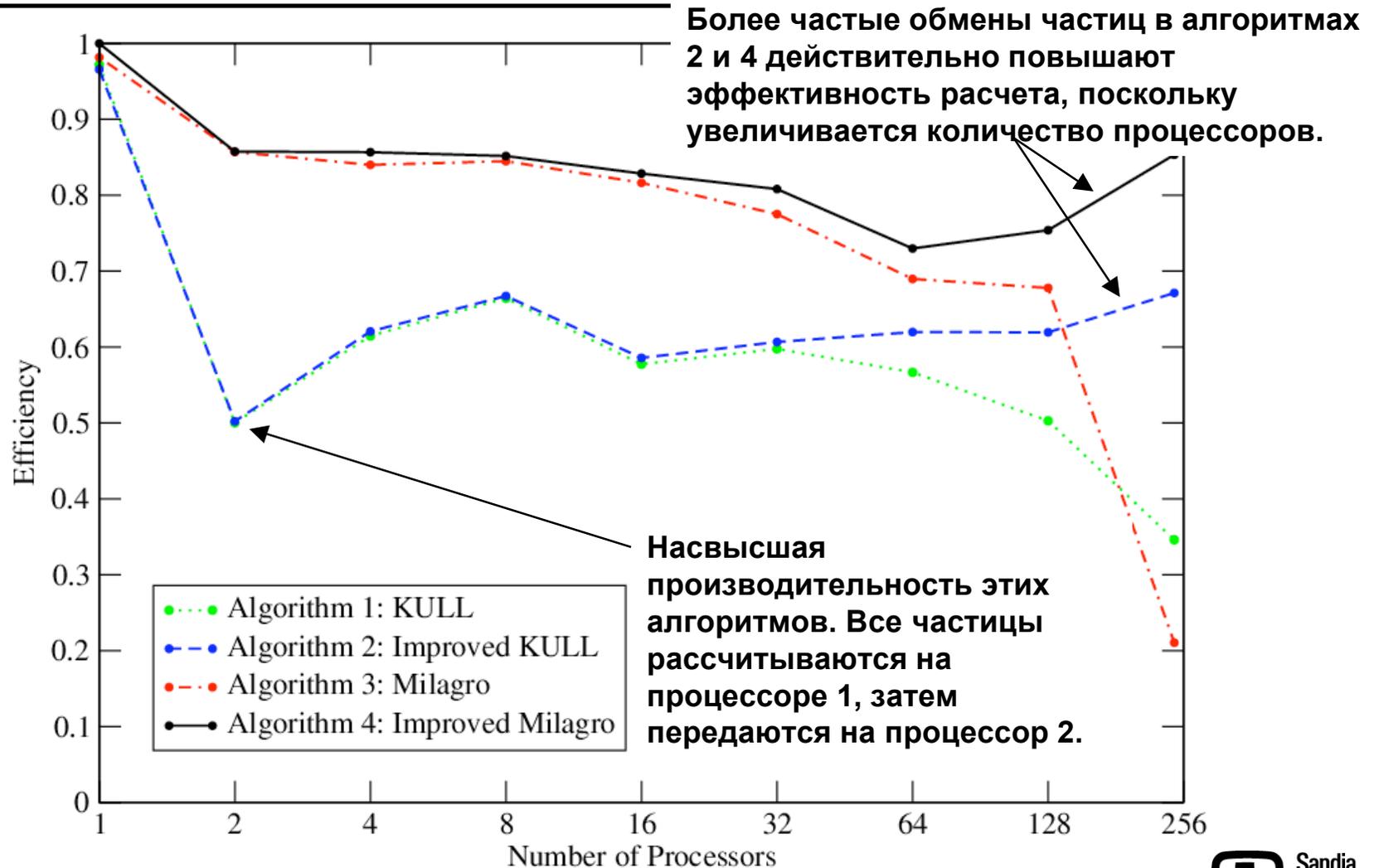
---

- Куб вакуума
- Отражающие границы
- Первоначальное холодное состояние
- С одной стороны находится источник температуры
- Начальный дисбаланс загрузки
- Загрузка балансируется к концу выполнения задачи





# Масштабирование постоянной работы для куба вакуума





## Выводы

---

- Улучшенный алгоритм Милагро, алгоритм 4, масштабируется почти линейно для задач со сбалансированной загрузкой.
- Очень важно не допускать в алгоритмах синхронизации всех точек, используя для этого либо блокирование передач, либо глобальные суммы.
- У каждого процессора должны быть одинаковые непроизводительные затраты при распараллеливании.
- Более частые обмены частицами могут помочь в решении задач с несбалансированной загрузкой.